**Processo de análise**

Esse serviço provê uma estrutura de aprovação, reprovação, suspender e reverter para uma etapa anterior. Além do mais, você poderá aplicar funções anônimas antes da execução do método principal (Ação que será executada) e também depois da execução.

Ele tem um conjunto de regras que podem ser definidas no escopo do serviço e você por padrão de estruturação, poderá criar uma classe com suas próprias regras.

1 – Inicialização da classe

Você precisará estender seu model da sua solicitação para o model (Servicos/Analyze/ProcessAnalyze).

Dentro dessa classe ele irá estender para o “Model” e você herdará todos os métodos do laravel.

$solicitation = new FinancyRPayment($payment, $request);  
$process = new ProcessAnalyze($solicitation);

// O eventStart é o método principal e o “rulesStatusStartAnalyze” está sendo ignorado. Você pode passar o nome de muitos métodos que queira ignorar do escopo global.

$result = $process->eventStart(['rulesStatusStartAnalyze']);

$result = $process->eventStart([], ['before' => ['rulesStatusStartAnalyze' => function($result) {

// Você pode pegar o resultado de regras globais antes mesmo de excutar.

}],

after => ['otherRules' => function($result, $result\_main) {

// o $result seria o resultado da regra e o $result\_main seria o resultado do método principal.

}],

]);

Echo $result;

2 – Criação da classe da solicitação

Você precisa respeita o processo de construção do serviço, criando uma classe para sua solicitação utilizando a interface do serviço de Analyze e implementando os métodos necessários para inicialização.

3 – Relação com as tabelas necessárias.

Você tem dois caminhos para decidir no momento de dar inicio análise de sua solicitação, você poderá trazer as pessoas que irão entrar no processo para aprovar de uma tabela intermediaria que você gravou as pessoas ou você poderá gravar diretamente na tabela mórfica sem informar o “analyze\_id”.

Sendo assim: Financy\_r\_payment é a tabela da minha solicitação, request\_analyze\_approvers é a tabela mórfica onde eu informo as pessoas que irão entrar na solicitação de aprovação.

Caso eu tivesse que de acordo com a situação da solicitação eu tivesse que mudar o grupo de pessoas que fosse aprovar, eu teria de informar o analyze\_id da tabela e as pessoas deverão estar dentro dessa tabela => request\_analyze\_approvers.

4 – Tabelas mórficas

Request\_analyze: Usada para mostrar o processo de anályze, ela conterá todas as pessoas que estão no processo de aprovação e nela será atualizado os status do processo.

Request\_analyze\_approv: Nessa tabela apenas guardará as pessoas que irão aprovar assim que iniciar o processo de análise. Obs: Se você depender de outra tabela para decidir as pessoas que irão aprovar, por favor, informe o analyze\_id da tabela e faça a relação mórfica, caso não tiver essa situação, informe apenas o namespace.

Request\_analyze\_observers: Os colaboradores que estiverem nessa tabela, receberão todas as informações da solicitação em cópia, conforme o processo for andando.

5 – Colunas

is\_approv, is\_reprov, is\_cancelled, cancel\_reason, cancel\_r\_code, has\_analyze, version

(INT) is\_approv = Usado para aprovar a solicitação.

(INT) is\_reprov = Usado para reprovar a solicitação.

(INT) has\_analyze = Mostra que a solicitação está em análise.

(INT) is\_cancelled = Indica que a solicitação foi cancelada.

(TEXT) cancel\_reason = A razão pelo qual cancelou a solicitação.

(VARCHAR(255)) cancel\_r\_code = Usuário que realizou o cancelamento da solicitação.

(INT Default(0)) version = Indica a versão em que está solicitação. As versões se atualização de acordo com a reprovação, mas em determinadas solicitações, pode se atualizada também depois que enviar novamente para aprovação depois mesmo de aprovado.